

Modeling multi-stage decision optimization problems

Ronald Hochreiter

April 2014

Abstract

Multi-stage optimization under uncertainty techniques can be used to solve long-term management problems. Although many optimization modeling language extensions as well as computational environments have been proposed, the acceptance of this technique is generally low, due to the inherent complexity of the modeling and solution process. In this paper a simplification to annotate multi-stage decision problems under uncertainty is presented - this simplification contrasts with the common approach to create an extension on top of an existing optimization modeling language. This leads to the definition of meta models, which can be instantiated in various programming languages. An example using the statistical computing language R is shown.

1 Introduction

We consider a multi-stage stochastic decision optimization framework based on a discrete-time decision process, i.e. there is a sequence of decisions at decision stages $t = 0, \dots, T$ where at each stage t a decision taker observes the realization of a random variable ξ_t , and takes a decision x_t based on all observed values ξ_0, \dots, ξ_t . At the terminal stage T a sequence of decisions $x = (x_0, \dots, x_T)$ with respective realizations $\xi = (\xi_0, \dots, \xi_T)$ leads to some cost $f(x, \xi)$. The goal is to find a sequence of decisions $x(\xi)$, which minimizes a functional of the cost $f(x(\xi), \xi)$. Multi-stage means that there is at least one intermediary stage between root stage and terminal stage.

The design goal of the approach presented in this paper is to design a modeling language independent of (a) the optimization modeling approach, e.g. expectation-based convex multi-stage stochastic programming or worst-case optimization, as well as (b) the underlying solution technique, e.g. either solving a scenario tree-based deterministic equivalent formulation or computing upper and lower bounds using primal and dual decision rules. Finally the modeling language should (c) be completely independent from a concrete programming language (C/C++, R, Matlab, Python, ...). The idea is to compose a meta model and instance concrete implementations semi-automatically.

Consider the two most common ways to solve multi-stage decision optimization problems, which is on one hand the scenario-based three-layered approach as shown in Fig. 1. See [9] for an overview of the area of stochastic programming, and [13] for stochastic programming languages, environments, and applications. More information on the modeling aspect can be found in [4].

The same decision problem may also be solved using scenario tree-free approximations [6] as shown in Fig. 2. The modeling language should be flexible enough to allow for applying any solution method, i.e. not being based on scenario trees, which is what most modeling language extensions for multi-stage models are proposing, see e.g. [10], [11], [7], [12], and [1].

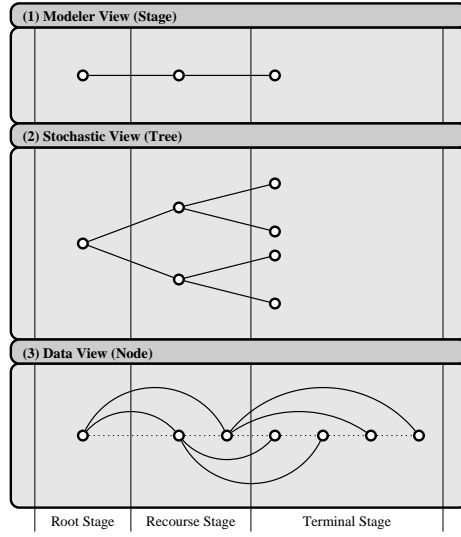


Figure 1: Scenario tree-based three-layered approach.

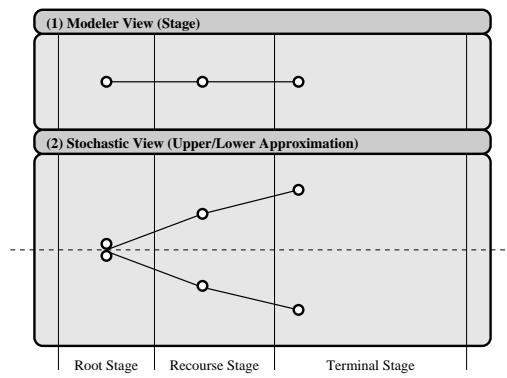


Figure 2: Scenario tree-free approximation.

2 Multi-stage Convex Stochastic Programming

Consider a multi-variate, multi-stage stochastic process ξ and a constraint-set \mathcal{X} defining a set of feasible combinations (x, ξ) . The set \mathcal{N} of functions $\xi \mapsto x$ are such that x_t is based on realizations up to stage t , i.e. only (ξ_0, \dots, ξ_t) . These are the non-anticipativity constraints. This leads to the general formulation shown in Equ. (1).

$$\begin{aligned} & \text{minimize } x : \mathbb{E}(f(x(\xi), \xi)) \\ & \text{subject to } (x(\xi), \xi) \in \mathcal{X} \\ & \quad x \in \mathcal{N} \end{aligned} \tag{1}$$

The most common way to solve such a problem is to create a scenario tree approximation of the underlying stochastic process and to build a deterministic equivalent formulation. The problem is that most modeling environments and languages are solely focussing on this type and mostly provide linear-only models due to solvability concerns. Furthermore, most allow for text-book applications only. There is almost no flexibility provided to extend models to use real-world objective functions and constraints.

The proposed solution is based on a complete decoupling of any scenario tree type of modeling from the decision problem modeling process, as shown in Fig. 1. On the decision problem (modeling) layer one should only be concerned with actions and decisions at stages. Other layers differ depending on the chosen solution method. In case of scenario trees and deterministic equivalent formulations there is an explicit decoupling of modeling and (scenario) tree handling, i.e. a scenario tree layer, whose focus is to create a scenario tree which optimally represents the subjective beliefs of the decision taker at each node. Furthermore there is an additional data layer, which handles the way how to (memory-)optimally store large scenario trees, access ancestor tree nodes quickly, and other computational (tree) operations.

3 Multi-stage Modeling Example

Consider the stylized simple multi-stage stochastic programming example from [3], which is shown in Equ. (2).

$$\begin{aligned} & \text{minimize } \mathbb{E}\left(\sum_{t=1}^T V_t x_t\right) \\ & \text{subject to } s_t - s_{t-1} = x_t \quad \forall t = 2, \dots, T \\ & \quad s_1 = 0, s_T = a, \\ & \quad x_t \geq 0, s_t \geq 0. \end{aligned} \tag{2}$$

The decision to be computed with this model is the optimal purchase over time under cost uncertainty, where the uncertain prices are given by V_t , and the decisions x_t are amounts to be purchased at each time period t . The objective function aims at minimizing expected costs such that a prescribed amount a is achieved at T ; s_t is a state variable containing the amount held at time t .

In Tab. 1 a concise meta formulation of this problem can be seen. The general syntax is borrowed from algebraic modeling languages like AMPL [2] and ZIMPL [5].

The most striking feature is that any relation to stages is removed from the definition of the optimization model - parameters, variables, objective function, and constraints. To accommodate for the definition of stages, the proposed stochastic modeling language contains two additional keywords for any of these objects, i.e.

Table 1: Modeling formulation of Equ. (2).

```

deterministic a: T;
stochastic x, s, objective_function: 0..T;
stochastic non_anticipativity: 1..T;
stochastic root_stage: 0;
stochastic terminal_stage: T;

param a;
var x >= 0, s >= 0;

minimize objective_function: E(V * x);
subject to non_anticipativity: s - s(-1) = x;
subject to root_stage: s = 0;
subject to terminal_stage: s = a;

```

- *deterministic objects: stage-set;*
- *stochastic objects: stage-set;*

Speaking in scenario tree notation the stochastic objects are defined on the underlying node structure and deterministic objects are defined on the stage structure, i.e. the latter contain the same value for all nodes in the respective stage. To define stochastic objective functions and stage recourse the following functions are defined, e.g. the most commonly used expectation functional for objective functions is simply expressed by the function $E()$. Furthermore, there is a special way to define stage-wise recourse for stochastic variables, i.e. `variable-name(recourse-depth)`. Note that while most modeling languages allow for a single stage recourse only, this definition allows for any number of recourse stages.

Tab. 2 shows the modeling example in some concrete implementation for the statistical computing language R [8]. This definition can be easily converted to a deterministic equivalent formulation or any other reformulation - all information is available in a concise format.

4 Conclusion

In this paper, a modeling language framework for a successful simplified meta modeling of multi-stage decision problems under uncertainty is shown, which allows for automatic reformulation and solution of multi-stage problems. This can be seen as a basis to build a model-based multi-stage problem library, especially because of its inherent decoupling from the underlying optimization technique as well as the fact that it is not bound to a specific programming language. Furthermore it is easy to integrate robust and stochastic optimization techniques to allow for comparing solutions to determine, which approach is optimally suited for which class of decision models. There are many ways to extend the proposed meta language - possible straight-forward extensions are e.g. quantiles for objective functions. In addition, application-related risk measures (shortcuts) can be defined, e.g. `CVaR(objects)`, as well as probabilistic constraints.

Table 2: Implementation of model 1 using the language R.

```

m <- model()
parameter(m, a)
variable(m, x, lb=0)
variable(m, s, lb=0)

minimize(m, "objective", "E(V * x)")
subject_to(m, "non_anticipativity", "s - s(-1) = x")
subject_to(m, "root_stage", "s = 0")
subject_to(m, "terminal_stage", "s = a")

deterministic(m, "T", a)
stochastic(m, "0..T", x, s, "objective")
stochastic(m, "1..T", "non_anticipativity")
stochastic(m, "0", "root_stage")
stochastic(m, "T", "terminal_stage")

optimize(m)

```

References

- [1] M. Colombo, A. Grothey, J. Hogg, K. Woodsend, and J. Gondzio. A structure-conveying modelling language for mathematical and stochastic programming. *Mathematical Programming Computation*, 1(4):223–247, 2009.
- [2] R. Fourer, D.M. Gay, and B.W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press/Brooks/Cole Publishing Company, 2nd edition, 2002.
- [3] H. Heitsch, W. Römisch, and C. Strugarek. Stability of multistage stochastic programs. *SIAM Journal on Optimization*, 17(2):511–525, 2006.
- [4] A.J. King and S.W. Wallace. *Modeling with Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer, 2013.
- [5] Thorsten Koch. *Rapid Mathematical Prototyping*. PhD thesis, Technische Universität Berlin, 2004.
- [6] D. Kuhn, W. Wiesemann, and A. Georghiou. Primal and dual linear decision rules in stochastic and robust optimization. *Mathematical Programming*, 130(1):177–209, 2011.
- [7] L. B. Lopes. *Modeling Stochastic Optimization: from Idea to Instance*. PhD thesis, Northwestern University, 2003.
- [8] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. URL <http://www.R-project.org>.
- [9] A. Ruszczyński and A. Shapiro, editors. *Stochastic programming*, volume 10 of *Handbooks in Operations Research and Management Science*. Elsevier Science B.V., Amsterdam, 2003.

- [10] J. Thénier, Ch. van Delft, and J.-Ph. Vial. Automatic formulation of stochastic programs via an algebraic modeling language. *Computational Management Science*, 4(1):17–40, 2007.
- [11] C. Valente, G. Mitra, M. Sadki, and R. Fourer. Extending algebraic modelling languages for stochastic programming. *INFORMS Journal on Computing*, 21(1):107–122, 2009.
- [12] Ch. van Delft and J.-Ph. Vial. A practical implementation of stochastic programming: an application to the evaluation of option contracts in supply chains. *Automatica*, 40(5):743–756, 2004.
- [13] S. W. Wallace and W. T. Ziemba, editors. *Applications of stochastic programming*, volume 5 of *MPS/SIAM Series on Optimization*. Society for Industrial and Applied Mathematics (SIAM), 2005.